In the Claims

1. (Currently Amended) A multiprocessor system comprising a first processor and a second processor, wherein:

the first processor ~~includes~~ comprises an interrupt generation unit which generates an interrupt to the second processor when the first processor executes a predetermined call instruction in a running main routine; and

the second processor ~~includes~~ comprises an address save unit which saves a return address for returning to the main routine upon completion of processing of a subroutine called by the call instruction to a predetermined memory area when the second processor receives an interrupt from the interrupt generation unit.

2. (Currently Amended) The multiprocessor system according to claim 1, wherein:

the interrupt generation unit generates an interrupt to the second processor again when a predetermined return instruction is executed in the subroutine; and

the second processor further ~~includes~~ comprises an address notification unit which notifies communicates the return address to the first processor when receiving the re-generated interrupt.

3. (Currently Amended) The multiprocessor system according to claim 1, wherein

the first processor ~~includes~~ comprises a fetcher which fetches an instruction; and the return address is set as a target address to be accessed by the fetcher.

4.(Currently Amended) A multiprocessor system comprising a first processor and a second processor, wherein:

the first processor ~~includes~~ comprises an interrupt generation unit which generates an interrupt to the second processor when the first processor executes a predetermined call instruction or jump instruction; and

the second processor ~~includes~~ comprises:

an address extraction unit which extracts a call destination address or jump destination address stored dividedly in formats of the call instruction or the jump instruction and an accompanying execution stop instruction when the second processor receives the interrupt from the first processor; and

an address notification unit which ~~notifies~~ communicates the acquired call destination address or jump destination address to the first processor.

5.(Currently Amended) The multiprocessor system according to claim 4, wherein:

the first processor ~~includes~~ further comprises a fetcher which fetches an instruction; and

the call destination address or the jump destination address is set as a target address to be accessed by the fetcher.

6. (Currently Amended) A multiprocessor system comprising a graphics processor and a main processor, wherein

the graphics processor ~~includes~~ comprises:

a direct memory access controller (DMAC) which reads instructions written in a display list from a memory ~~in succession~~ sequentially;

a decoder which decodes the read instructions ~~in succession~~ sequentially; and

an interrupt generation unit which generates a shift interrupt to the main processor when a decoded instruction is a predetermined call instruction included in a main routine of the display list and generates a return interrupt to the main processor when a decoded instruction is a return instruction included in a subroutine called by the call instruction,

the main processor ~~includes~~ comprises:

an address save unit which saves a return address for returning to the main routine upon completion of processing of the subroutine to a predetermined memory when the main processor receives the shift interrupt from the interrupt generation unit; and

an address notification unit which reads the return address from the predetermined memory and ~~notifies~~ communicates the return address to the graphics processor when the main processor receives the return interrupt from the interrupt generation unit, and

the return address ~~notifies~~ communicated to the graphics processor is set as a target address to be accessed by the DMAC.

7.(Currently Amended) A multiprocessor system comprising a graphics processor and a main processor, wherein

the graphics processor ~~includes~~ comprises:

a direct memory access controller (DMAC)which reads instructions written in a display list from a memory ~~succession~~ sequentially;

a decoder which decodes the read instructions ~~succession~~ sequentially; and

an interrupt generation unit which generates an interrupt to the main processor when a decoded instruction is a predetermined call instruction or a jump instruction included in the display list,

the main processor ~~includes~~ comprises an address notification unit which acquires a call destination address or a jump destination address stored dividedly in formats of the call instruction or the jump instruction and an accompanying execution stop instruction, and ~~notifies~~ communicates the call destination address or the jump destination address to the graphics processor when the main processor receives the interrupt from the interrupt generation unit, and

the call destination address or the jump destination address ~~notified~~ communicated to the graphics processor is set as a target address to be accessed by the DMAC.

8.(Currently Amended) A method of executing a program in a multiprocessor system, the method comprising, ~~when a first processor executes~~ executing a call instruction in a ~~running~~ main routine running by a first processor, ~~committing~~ delegating to a second processor the task of saving ~~save of~~ a return address for returning to the main routine upon completion of processing of a subroutine called by the call instruction ~~to a second processor~~.

9.(Currently Amended) The method of executing a program in a multiprocessor system according to claim 8, wherein:

if a stack area inside the first processor has a free space, the first processor saves the return address to the stack area by itself; and

if the stack area has no free space, the save of the return address is ~~committed~~ delegated to the second processor.

10.(Currently Amended) The method of executing a program in a multiprocessor system according to claim 8, wherein:

if the call instruction does not explicitly instruct to delegate ~~commit~~ the task of saving ~~save of~~ the return address to the second processor, the first processor saves the return address to a stack area built in itself; and

if the call instruction explicitly instructs to ~~commit the save of~~ delegate the task of saving the return address to the second processor, the first processor ~~commits the save of~~ delegates the task of saving the return address to the second processor.

11. (Currently Amended) A method of executing a program in a multiprocessor system, the method comprising, executing a call instruction or a jump instruction by a first processor, delegating a task of acquiring a full address of a call destination address or a jump destination address to a second processor ~~if the number of bits of the call destination address or jump destination address exceeds the number of bits acquirable by the first processor~~.

12. (Currently Amended) A method of executing a program in a multiprocessor system according to claim 11, wherein ~~the method comprising, executing a call instruction or a jump instruction by a first processor,~~ delegating a task of acquiring the full address of the ~~a~~ call destination address or the jump destination address to a second processor, if the number of bits of ~~if the call instruction or the jump instruction explicitly instructs to delegate the task of acquiring~~ the call destination address or the jump destination address exceeds the number of bits acquirable by the first processor ~~to the second processor~~.

13. (Previously Canceled)

14. (New) A method of executing a program in a multiprocessor system according to claim 11, wherein delegating the task of acquiring the full address of the call destination address or the jump destination address to a second processor if the call instruction or the jump instruction explicitly instructs to delegate the task of acquiring the call destination address or the jump destination address to the second processor.